

Facultad: Ingeniería.
Escuela: Electrónica.
Asignatura: Microprocesadores.
Lugar de ejecución: Laboratorio de
Automatización (Edificio 3, 1a planta).

Introducción al uso de la tarjeta de entrenamiento DEMO9S12XDT512

Objetivos específicos

- Realizar un programa en ensamblador para procesador 9S12XDT512
- Depurar el programa con el simulador
- Descargar el programa a la memoria por medio del CodeWarrior IDE

Materiales y equipo

- 1 Computadora con Windows 98 o superior con el software CodeWarrior IDE
- 1 Tarjeta DEMO9S12XDT512
- 1 Cable de conexión USB.

Procedimiento

Creación del Programa en CodeWarrior IDE

1. Abra el programa CodeWarrior Development Studio.
2. Cree un nuevo proyecto con el nombre de su preferencia por medio del asistente para la creación de proyectos.
3. Seleccione como derivativo a utilizar como MC9S12XDT512
4. Para el soporte de XGATE seleccione Single Core
5. Para el lenguaje seleccione solamente la opción Assembly
6. En el tipo de ensamblaje seleccione la opción Absolute Assembly
7. Para las conexiones seleccione Full Chip Simulation

El Mapa de memoria de la tarjeta es el siguiente:

\$0000 - \$07FF	REGISTERS	2 KB	
\$0800 - \$0FFF	EEPROM	4 KB	4 – 1Kb pages between 0x800 – 0xBFF
\$1000 - \$3FFF	RAM	12 KB	5 – 4Kb pages between 0x1000 – 0x1FFF
\$4000 - \$7FFF	FIXED FLASH	16 KB	1K, 2K, 4K, 8K Protected Boot Sector
\$8000 - \$BFFF	FLASH EEPROM PAGE WINDOW	16 KB	32 – 16Kb pages
\$C000 - \$EFFF	FIXED FLASH	16 KB	2K, 4K, 8K, 16K Protected Boot Sector
\$FF00 - \$FFFF	Vectors BDM (if active)	255 bits	

Tabla 1. Mapa de memoria de la tarjeta DEMO9S12XDT512

En esta guía se hará mención de puertos de entrada y salida, puede por el momento pensar en ellos como localidades de memoria que pueden ser cambiados por el usuario, en próximas guías se realizarán aplicaciones para su uso.

8. Modifique el archivo main.asm que deberá quedar con su texto de la siguiente manera:

```

;*****
;* This stationery serves as the framework for a *
;* user application (single file, absolute assembly application) *
;* For a more comprehensive program that *
;* demonstrates the more advanced functionality of this *
;* processor, please see the demonstration applications *
;* located in the examples subdirectory of the *
;* Freescale CodeWarrior for the HC12 Program directory *
;*****
; export symbols
        XDEF Entry          ; export 'Entry' symbol
        ABSENTRY Entry    ; for absolute assembly: mark this as application entry point

; Include derivative-specific definitions
        INCLUDE 'derivative.inc'
ROMStart EQU $4000 ; absolute address to place my code/constant data
; variable/data section
        ORG $1000
; Insert here your data definition.
COMBINACION DS.B 1
; code section

```

```

        ORG    ROMStart
Entry:
        LDS    #RAMEnd+1      ; initialize the stack pointer
        CLI                      ; enable interrupts
        LDAA  #$F0
        STAA  DDRB            ;Pone 4 bits más significativos como salida
                                ;y 4 bits menos significativos como entrada
        LDAA  #$0F0
        STAA  PORTB          ;Inicializa los leds en la parte más significativa

        BSET  PUCR,2         ;Habilita las resistencias de PULL UP del Puerto B
Inicio:
        LDAA  COMBINACION      ;Carga el dato de 4 bits a comparar
                                ; Se cambiará luego a PORTB
        ANDA  #$0F            ;Enmascara 4 bits menos significativos
        CMPA  #$05            ;Combinación 0110
        BEQ  ActivaLED1
        CMPA  #$09            ;Combinación 1001
        BEQ  ActivaLED2
        CMPA  #$0A            ;Combinación 1010
        BEQ  ActivaLED3
        CMPA  #$0D            ;Combinación 1101
        BEQ  ActivaLED4
        BRA  Inicio

ActivaLED1: LDAB  #$E0
            STAB  PORTB
            BRA  Inicio
ActivaLED2: LDAB  #$D0
            STAB  PORTB
            BRA  Inicio
ActivaLED3: LDAB  #$B0
            STAB  PORTB
            BRA  Inicio
ActivaLED4: LDAB  #$70
            STAB  PORTB
            BRA  Inicio
;*****
;*          Interrupt Vectors          *
;*****
        ORG    $FFFE
        DC.W  Entry                ; Reset Vector

```

Listado 1. Programa de puertos

9. Simule el comportamiento del programa seleccionando la opción del Menú Project /Debug o con la tecla de Función F5.
10. Para simular el cambio en la memoria solamente cambie la dirección \$1000 en la ventana que dice memory dando clic derecho y eligiendo la opción Address del menú

contextual tal como se muestra en la siguiente figura:

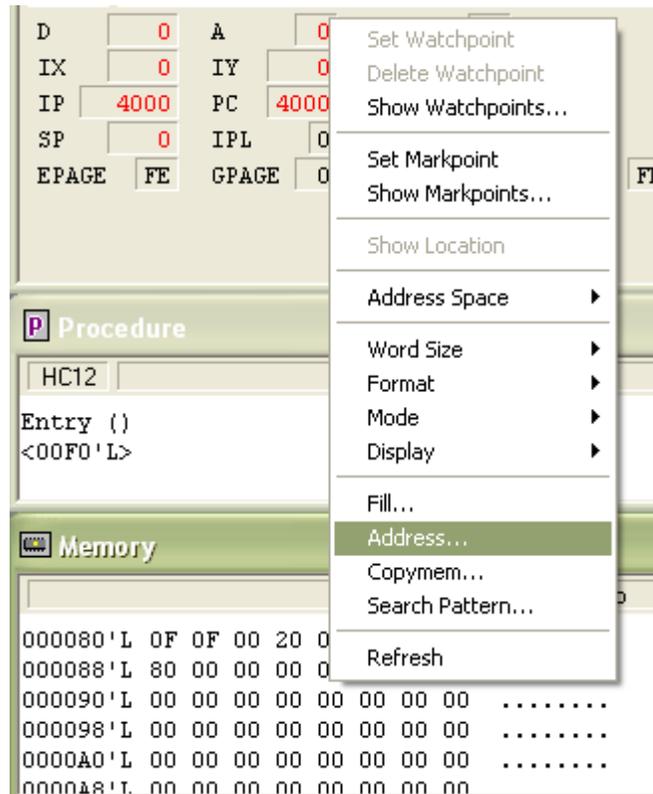


Figura 1 Elegir localidad de memoria a editar

11. Aparecerá luego otro cuadro de diálogo donde tendrá que introducir el valor de la localidad deseada ver figura 2.



Figura 2 Desplegar Dirección

12. Cambie el cero que aparece al inicio por 1000, deje el apóstrofe y la L
13. En la ventana de memoria aparecerá resaltada la localidad deseada, dé doble clic sobre esta y cámbiela por 05.
14. Corra el programa paso a paso hasta llegar al salto BEQ ActivaLED1, observe después de la comparación con el dato 05 hacia adónde se dirige el programa en el siguiente paso, siga paso a paso hasta llegar de nuevo a Inicio.
15. Cambie la dirección de memoria de COMBINACION por el dato 09 y observe si se cumple la condición del segundo salto.
16. Observe si se cumplen las otras dos condiciones para los datos 0A y 0D
17. Cambie en el archivo asm la línea LDAA COMBINACION, sustituya la palabra COMBINACION, por la palabra PORTB, que hace alusión al puerto físico B, en sus 4

bits menos significativos tiene conectados 4 Switches que son los que harán la combinación para datos de entrada y en sus 4 bits más significativos se tienen 4 leds conectados y son activos en bajo (con 0 en la línea del puerto el led enciende).

18. Conecte la tarjeta DEMO9S12XDT512 al puerto usb y se realizará la descarga del programa. Para esto se requiere establecer una conexión con la tarjeta, seleccione en el simulador la opción Component / Set Connection. Seleccione las opciones presentes en la siguiente figura

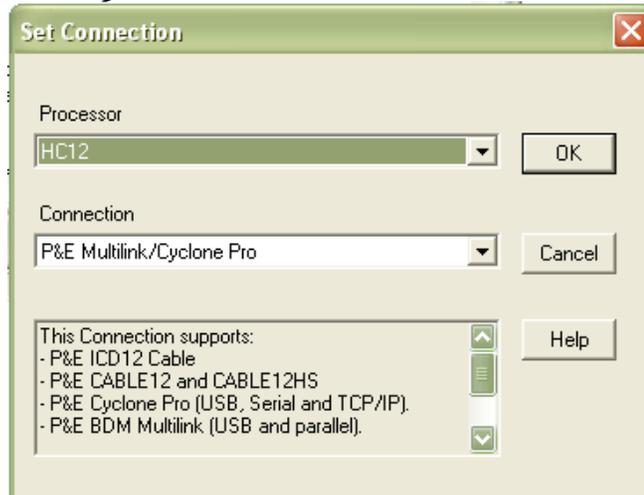


Figura 3 Opciones para la conexión con la tarjeta DEMO9S12XDT512

19. Al haber ejecutado la conexión con la tarjeta habrá que cargar el archivo ya compilado a la memoria de la tarjeta de evaluación, para esto observe que se ha cargado un menú que dice MultilinkCyclonePro, seleccione la opción Load como se muestra en la siguiente figura



Figura 4. Cargar archivo binario a memoria de la tarjeta de evaluación.

20. Se abrirá un cuadro de diálogo que tendrá las carpetas de su proyecto seleccione de la carpeta bin el archivo con extensión “.abs” y luego abrir, se comenzará a realizar el

proceso de descarga del programa, cuando se dé un mensaje de advertencia del borrado de la memoria acepte dando clic en OK.

21. Ejecute el programa dando clic en el botón de la ventana de depuración  y verifique el correcto funcionamiento cambiando los switches presentes en la tarjeta por las respectivas combinaciones 05, 09, 0A y 0D y observando la respuesta de los leds.
22. Ejecute el programa paso a paso y viendo las direcciones de memoria correspondientes y el resultado en los leds.
23. Haga unos cambios al código en el archivo de texto según el siguiente listado

```

;*****
;* This stationery serves as the framework for a
;* user application (single file, absolute assembly application)
;* For a more comprehensive program that
;* demonstrates the more advanced functionality of this
;* processor, please see the demonstration applications
;* located in the examples subdirectory of the
;* Freescale CodeWarrior for the HC12 Program directory
;*****

; export symbols
        XDEF Entry          ; export 'Entry' symbol
        ABSENTRY Entry ; for absolute assembly: mark this as application entry point
; Include derivative-specific definitions
        INCLUDE 'derivative.inc'
ROMStart EQU $4000 ; absolute address to place my code/constant data
; variable/data section
        ORG $1000
; Insert here your data definition.
COMBINACION DS.B 1
LEDS dc.b $E0,$D0,$B0,$70
; code section
        ORG ROMStart
Entry:
        LDS #RAMEnd+1      ; initialize the stack pointer
        CLI                ; enable interrupts
        LDAA #$F0
        STAA DDRB          ;Pone 4 bits más significativos como salida
                           ;y 4 bits menos significativos como entrada
        LDAA #$0F0
        STAA PORTB        ;Inicializa los leds en la parte más significativa

        BSET PUCR,2 ;Habilita las resistencias de PULL UP del Puerto B
Inicio:
        LDAA PORTB        ;Carga el dato de 4 bits a comparar
                           ; Se cambiará luego a PORTB
        ANDA #$0F
        CMPA #$05        ;Combinación 0110

```

```

        BEQ ActivaLED1
        CMPA  #$09           ;Combinación 1001
        BEQ ActivaLED2
        CMPA  #$0A           ;Combinación 1010
        BEQ ActivaLED3
        CMPA  #$0D           ;Combinación 1101
        BEQ ActivaLED4
        BRA  Inicio

ActivaLED1:LDX  #$0000
            BRA  Activar
ActivaLED2:LDX  #$0001
            BRA  Activar
ActivaLED3:LDX  #$0002
            BRA  Activar
ActivaLED4:LDX  #$0003

Activar:   LDAB  LEDS,X
            STAB  PORTB
            BRA  Inicio
;*****
;*                Interrupt Vectors                *
;*****
        ORG   $FFFE
        DC.W  Entry           ; Reset Vector

```

Listado 2. Manejo de vectores de constantes

24. Compruebe el programa
25. Explique el funcionamiento
26. Cierre el depurador
27. Accione el botón reset de la tarjeta de evaluación y verifique si el programa se ejecuta de forma automática aun sin tener el depurador abierto.
28. Desconecte la tarjeta, cierre el programa CodeWarrior IDE y apague la computadora

Análisis de Resultados

- Verifique los símbolos utilizados en el programa y (PUCR, PORTB, DDRB) qué representan y en qué direcciones pueden localizarse.

Bibliografía

- Manual de la tarjeta de Evaluación DEMO9S12XDT512
- Manual de Instrucciones hc12
- Ayuda de programa CodeWarrior IDE